

華東理工大學

畢業論文(設計)

題 目	<u>狗屁不通文章生成器</u> <u>但是把标题加长到两行</u>
学 院	<u>复制粘贴写报告学院</u>
系 别	<u></u>
专 业	<u></u>
年 级	<u></u>
学 号	<u>U000114514</u>
姓 名	<u>作者</u>
导 师	<u>你的老板</u>

定稿日期： 1926 年 8 月 17 日

華東理工大學

毕业设计(论文)任务书

题 目 狗屁不通文章生成器

但是把标题加长到两行

2024 年 12 月 23 日——2025 年 6 月 8 日，共计 18 周

院 系 信息科学与工程学院
电子与通信工程系

专业班级

学 号

姓 名

教 学 院 长

系 主 任

指 导 教 师

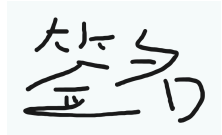
2024 年 12 月 23 日

课题来源	
课题的目的、意义	
要求	
课题主要内容及进度	<p>本课题采用 RAG 增强生成技术，结合 Agent 的自主决策能力，实现知识库构建，RAG 模型优化，Agent 模块功能。</p> <p>进度安排：</p> <p>1~2 周 查阅 RAG 与教育 AI 领域文献，完成关键技术综述翻译，搭建开发环境；</p> <p>3~4 周 完善文献综述，撰写开题报告，确定系统架构与技术选型；</p> <p>5~6 周 分析教育知识库构建流程，完成数据采集与预处理（爬虫/OCR）；</p> <p>7~9 周 实现 RAG 核心模块：向量化检索、生成模型微调、Agent 决策逻辑开发；</p> <p>10~13 周 系统集成与多场景测试（如课件生成、智能答疑、学习路径规划）；</p> <p>14~16 周 撰写论文，优化实验指标；</p> <p>17~18 周 制作 PPT、正式答辩与系统演示。</p>

作 者 声 明

我郑重声明：本人恪守学术道德，崇尚严谨学风。所呈交的毕业论文（设计），是本人在导师的指导下，独立开展工作所取得的结果。除文中明确注明和引用的内容外，本论文（设计）不包含任何他人已经发表或撰写过的内容。论文（设计）为本人亲自完成，并对所写内容负责。

论文（设计）作者签名：



2025 年 5 月 16 日

摘要

先帝创业未半而中道崩殂，今天下三分，益州疲弊，此诚危急存亡之秋也。然侍卫之臣不懈于内，忠志之士忘身于外者，盖追先帝之殊遇，欲报之于陛下也。诚宜开张圣听，以光先帝遗德，恢弘志士之气，不宜妄自菲薄，引喻失义，以塞忠谏之路也。

宫中府中，俱为一体；陟罚臧否，不宜异同。若有作奸犯科及为忠善者，宜付有司论其刑赏，以昭陛下平明之理，不宜偏私，使内外异法也。

关键词： 关键词 1，关键词 2，关键词 3

Abstract

The founding emperor passed away before his endeavor was half completed, and now the empire is divided into three parts. Yizhou is exhausted and in decline, and this is truly a critical moment of survival or destruction. However, the palace guards are tirelessly serving within, and loyal subjects are sacrificing themselves outside, all in order to repay the late emperor's kindness and show loyalty to the current emperor. It is appropriate to listen to wise advice, to honor the late emperor's virtues, to inspire the courage of loyal subjects, and not to belittle oneself or distort the truth, in order to keep the path of loyal counsel open.

The palace and government are one entity, and punishments should be consistent. If there are those who commit crimes or show loyalty and virtue, they should be judged by the legal system to demonstrate your fairness as emperor, rather than showing partiality that would create different laws for those inside and outside the palace.

Keywords: Keyword 1,Keyword 2,Keyword 3

目录

摘要	I
Abstract	II
1 绪论	1
1.1 typst 介绍	1
1.2 基本语法	1
1.2.1 代码执行	1
1.2.2 标题	1
2 本模板相关封装	2
2.1 图片	2
2.2 表格	2
2.3 公式	3
2.4 列表	4
2.5 代码	4
2.5.1 代码块	4
2.5.2 伪代码	6
3 其他说明	7
3.1 文献引用	7
3.1.1 数据相关	7
4 这是一章占位的	8
4.1 占位的二级标题 1	8
4.2 占位的二级标题 2	8
4.3 占位的二级标题 3	8
4.4 占位的二级标题 4	8
4.4.1 占位的三级标题 1	8
4.4.2 占位的三级标题 2	8
4.5 占位的二级标题 5	8
4.6 占位的二级标题 6	8

参考文献	9
附录 A	10
附录 B	11
致谢	12

1 绪论

1.1 typst 介绍

typst 是最新最热的标记文本语言，定位与 LaTeX 类似，具有极强的排版能力，通过一定的语法写文档，然后生成 pdf 文件。与 LaTeX 相比有以下的优势：

1. 编译巨快：因为提供增量编译的功能所以在修改后基本能在一秒内编译出 pdf 文件，typst 提供了监听修改自动编译的功能，可以像 Markdown 一样边写边看效果。
2. 环境搭建简单：原生支持中日韩等非拉丁语言，不用再大量折腾字符兼容问题以及下载好几个 G 的环境。只需要下载命令程序就能开始编译生成 pdf。
3. 语法友好：对于普通的排版需求，上手难度跟 Markdown 相当，同时文本源码阅读性高：不会再充斥一堆反斜杠跟花括号

个人观点：跟 Markdown 一样好用，跟 LaTeX 一样强大



图 1-1 我的 image 实例 0

1.2 基本语法

1.2.1 代码执行

正文可以像前面那样直接写出来，隔行相当于分段。

个人理解：typst 有两种环境，代码和内容，在代码的环境中会按代码去执行，在内容环境中会解析成普通的文本，代码环境用 `{}` 表示，内容环境用 `[]` 表示，在 `content` 中以 `#` 开头来接上一段代码，比如 `#set rule`，而在花括号包裹的块中调用代码就不需要 `#`。

1.2.2 标题

类似 Markdown 里用 `#` 表示标题，typst 里用 `=` 表示标题，一级标题用一个 `=`，二级标题用两个 `=`，以此类推。

间距、字体等会自动排版。

`#pagebreak()` 函数相当于分页符。论文里第一级标题应当分页，本模板已配置自动分页。

2 本模板相关封装

2.1 图片

内部用模块 `i-figure` 实现图片，表格与公式的编号。



图 2-1 我的 `image` 实例 1

引用的话就在 `figure` 后加上标签，在原标签前加上前缀 `fig:`, `tbl:`, `eqt:`, `lst:`（分别对应图片，表格、公式与代码块）。



图 2-2 我的 `image` 实例 2

用 `@fig:img1` 引用 2-1: 图 2-1

2.2 表格

表格跟图片差不多，但是表格的输入要复杂一点，建议在 `typst` 官网学习，自由度特别高，定制化很强。

同样使用 `figure` 函数包裹表格

表 2.1 芝士样表



	Area	Parameters
	$\pi h \frac{D^2 - d^2}{4}$	h : height D : outer radius d : inner radius
	$\frac{\sqrt{2}}{12} a^3$	a : 边长

表 2.2 三线表示例

Country List	Country List	Country List
Country Name or Area Name	ISO ALPHA Code	ISO ALPHA
Afghanistan	AF	AFT
Aland Islands	AX	ALA
Albania	AL	ALB
Algeria	DZ	DZA
American Samoa	AS	ASM
Andorra	AD	AND
Angola	AP	AGO

表 2.3 三线表示例 2

Country List	Country List	Country List	Country List
Country Name or Area Name	ISO ALPHA 2 Code	ISO ALPHA 3	8
Afghanistan	AF	AFT	7
Aland Islands	AX	ALA	6
Albania	AL	ALB	5
Algeria	DZ	DZA	4
American Samoa	AS	ASM	3
Andorra	AD	AND	2
Angola	AP	AGO	1

2.3 公式

公式用两个 \$ 包裹，但是语法跟 LaTeX 并不一样，如果有大量公式需求建议看官网教程 <https://typst.app/docs/reference/math/equation/>。

不需要封装即可为公式编号

$$A = \pi r^2 \quad (2-1)$$

根据公式 (2-1)，推断出公式 (2-2)

$$x < y \Rightarrow x \not\geq y \quad (2-2)$$

然后也有多行的如公式 (2-3)，标签名字可以自定义

$$\sum_{k=0}^n k < 1 + \dots + n \quad (2-3)$$
$$= \frac{n(n+1)}{2}$$

如果不想编号就在公式后面使用标签 <->

$$x < y \Rightarrow x \not\geq y$$

当然，你也可以使用 `mitex` 包直接渲染 `latex` 公式，这在你的公式已经是 `latex` 格式的情况下（例如 `simpletex` 识别结果等）很好用。

2.4 列表

- 无序列表 1: 1
- 无序列表 2: 2

列表后的正文，应当有缩进。这里加入一个 `#indent()` 函数来手动生成段落缩进，是因为在目前的 `typst` 设计里，按英文排版的习惯，连续段落里的第一段不会缩进，也包括各种列表。

1. 有序列表 1
2. 有序列表 2

列表后的正文，应当有缩进，但是这里没有，请自己在段首加上 `#indent()`

想自己定义可以自己 `set numbering`，建议用 `#[]` 包起来保证只在该作用域内生效：

- 1) 自定义列表 1
 - a) 自定义列表 1.1
- 2) 自定义列表 2
 - a) 自定义列表 2.1

2.5 代码

2.5.1 代码块

代码块使用的是库 `codelst`。

```
1 #show "ArtosFlow": name => box[
2   #box(image(
3     "logo.svg",
4     height: 0.7em,
5   ))
6   #name
```

```
7 ]  
8  
9 This report is embedded in the  
10 ArtosFlow project. ArtosFlow is a  
11 project of the Artos Institute.
```

如果你需要为代码块添加 `caption`，可以像下面这样：

```
1 #figure(  
2     sourcecode[...],  
3     kind: raw,  
4     caption: "代码块caption示例",  
5 )
```

代码 2-1 代码块 `caption`

但是一个很长的带有 `figure` 的代码块无法跨页，此时需要用下面的形式。

```
1 一个很  
2 长  
3 长  
4 长  
5 长  
6 长  
7 长  
8 长  
9 长  
10 长  
11 长  
12 长  
13 长  
14 长  
15 长  
16 长  
17 长  
18 长  
19 长  
20 长  
21 长  
22 长  
23 长  
24 长  
25 长  
26 长  
27 长  
28 长  
29 长  
30 长  
31 长  
32 长  
33 长  
34 长  
35 长  
36 长  
37 长  
38 长  
39 长  
40 长  
41 长
```

```

42 长
43 长
44 长
45 长
46 长
47 长
48 长
49 长
50 的代码块

```

代码 2-2 代码块 caption

(比较奇怪的是在全局设置 `#show figure.where(kind: raw): set block(breakable: true)` 是无效的，所以需要这样让用户手动操作，希望有其他 typst 佬解惑)

2.5.2 伪代码

计算机学院建议使用如下的伪代码模版进行伪代码书写。

reference link: <https://typst.app/universe/package/lovelace>

```

input: integers  $a$  and  $b$ 
output: greatest common divisor of  $a$  and  $b$ 
1      while  $a \neq b$  do
2          if  $a > b$  then
3               $a \leftarrow a - b$ 
4          else
5               $b \leftarrow b - a$ 
6          end
7      end
8      return  $a$ 

```

算法 2-1 The Euclidean algorithm

3 其他说明

3.1 文献引用

引用支持 LaTeX Bib 的格式。在引用时使用 `@tag`，像这样^[1,2] 以获得右上的引用标注^[3,4]。

记得在最后加入 `#references("path/to/ref.bib")` 函数的调用来生成参考文献。

3.1.1 数据相关

所有拉丁字母均为 Times New Roman，大小与汉字相同

正文为宋体小四

图表标题为黑体小四

表格单元格内容宋体五号

一级标题黑体小二加粗

二级标题四号加粗

正文行间距 0.95em（肉眼测量）

a4 纸，上空 2.8cm，其他空 2.5cm

4 这是一章占位的

4.1 占位的二级标题 1

4.2 占位的二级标题 2

4.3 占位的二级标题 3

4.4 占位的二级标题 4

4.4.1 占位的三级标题 1

4.4.2 占位的三级标题 2

4.5 占位的二级标题 5

4.6 占位的二级标题 6

参考文献

- [1] IMPAGLIAZZO R, PATURI R, ZANE F. Which problems have strongly exponential complexity?[J]. Journal of Computer and System Sciences, 2001, 63(4): 512-530.
- [2] BURCKHARDT S, FAHNDRICH M, HALLEUX P de, et al. It's Alive! Continuous Feedback in UI Programming[J/OL]. SIGPLAN Not., 2013, 48(6): 95-104. <https://doi.org/10.1145/2499370.2462170>. DOI:10.1145/2499370.2462170.
- [3] 刘星, 唐勇. 恶意代码的函数调用图相似性分析[J]. 计算机工程与科学, 2014, 36(3): 481-486.
- [4] PAPADIMITRIOU C H, STEIGLITZ K. Combinatorial optimization: algorithms and complexity[M]. Courier Corporation, 1998.

附录 A

使用 `<nonumber>` 标签将标题设为无编号。在写附录和致谢时很有用。

附录 B

这里是附录 B 内容。

致谢

完成本篇论文之际，我要向许多人表达我的感激之情。

首先，我要感谢我的指导教师，他/她对本文提供的宝贵建议和指导。所有这些支持和指导都是无私的，而且让我受益匪浅。

其次，我还要感谢我的家人和朋友们，他们一直以来都是我的支持和鼓励者。他们从未停止鼓舞我，勉励我继续前行，感谢你们一直在我身边，给我幸福和力量。

此外，我还要感谢我的同学们，大家一起度过了很长时间的时光，互相支持和鼓励，共同进步。因为有你们的支持，我才能不断地成长、进步。

最后，我想感谢笔者各位，你们的阅读和评价对我非常重要，这也让我意识到了自己写作方面的不足，同时更加明白了自己的研究方向。谢谢大家！

再次向所有支持和鼓励我的人表达我的谢意和感激之情。

本致谢生成自 ChatGPT。